Our Ref.: 42390.P7948

APPLICATION FOR UNITED STATES LETTERS PATENT

FOR

Method And Apparatus For Protected Exchange Of Status And Secret Values Between A Video Source Application And A Video Hardware Interface

Inventor(s): **Robert W. Faber**
David A. Lee
Brendan S. Traw
Gary L. Graunke
Richard P. Mangold

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN, LLP
12400 Wilshire Boulevard, 7th Floor
Los Angeles, California 90025
(503) 684-6200

"Express Mail" Label Number *EL431686001US*

Method And Apparatus For Prot ct d Exchang Of Status And Secr t Values B twe n A Vid o Source Application and A Vid o Hardware Interface

Related Application

This application is a continuation-in-part application to U.S. Patent Applications number 09/385,590 and 09/385,592, both entitled Digital Video Content Transmission Ciphering and Deciphering Method and Apparatus, filed on August 29, 1999.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to the field of content protection. More specifically, the present invention addresses the protection accorded to exchange of status and secret values between a video source application and a video hardware interface of a video source device.

2. Background Information

In general, entertainment, education, art, and so forth (hereinafter collectively referred to as "content") packaged in digital form offer higher audio and video quality than their analog counterparts. However, content producers, especially those in the entertainment industry, are still reluctant in totally embracing the digital form. The primary reason being digital contents are particularly vulnerable to pirating. As unlike the analog form, where some amount of quality degradation generally occurs with each copying, a pirated copy of digital content is virtually as good as the "gold master". As a result, much effort have been spent by the industry in developing and

adopting techniques to provide protection to the distribution and rendering of digital content.

Historically, the communication interface between a video source device (such as a personal computer) and a video sink device (such as a monitor) is an analog interface. Thus, very little focus has been given to providing protection for the transmission between the source and sink devices. With advances in integrated circuit and other related technologies, a new type of digital interface between video source and sink devices is emerging. The availability of this type of new digital interface presents yet another new challenge to protecting digital video content. While in general, there is a large body of cipher technology known, the operating characteristics such as the volume of the data, its streaming nature, the bit rate and so forth, as well as the location of intelligence, typically in the source device and not the sink device, present a unique set of challenges, requiring a new and novel solution. Parent applications number 09/385,590 and 09/385,592 disclosed various protocol and cipher/deciphering techniques to protect the transmission.

Similar protection challenges exist for exchanges of status and secret values between the video generating video source application and the video transmitting video hardware interface of the video source device. Thus, method and apparatus to protect these exchanges are desired.

SUMMARY OF THE INVENTION

A video source application in a video source device requests from a video hardware interface of the video source device status with respect to a link linking the video source device to an external video sink device, and supplements the status request with a basis value to a symmetric ciphering/deciphering process. The video source application, upon receiving from the video hardware interface the requested status and a verification key, generated using a symmetric ciphering/deciphering process and employing the basis value, verifies the correctness of the verification key to determine whether to trust said provided status.

BRIEF DESCRIPTION OF DRAWINGS

The present invention will be described by way of exemplary embodiments, but not limitations, illustrated in the accompanying drawings in which like references denote similar elements, and in which:

Figure 1 illustrates an overview of the present invention in accordance with one embodiment;

Figures 2a-2b illustrate a symmetric ciphering/deciphering process based method for the video hardware interface to provide sensitive information such as status and secret values to the video source application, in accordance with two embodiments;

Figures 3a-3b illustrate the symmetric ciphering/deciphering process of **Fig. 2a-2b** employed to facilitate provision of status and secret values from the video hardware interface to the video source application, in accordance with one embodiment each; and

Figures 4a-4c illustrate a one way function suitable for use to practice the symmetric ciphering/deciphering process of **Fig. 3a-3b** in further detail, in accordance with one embodiment.

DETAILED DESCRIPTION OF THE INVENTION

In the following description, various aspects of the present invention will be described, and various details will be set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to those skilled in the art that the present invention may be practiced with only some or all aspects of the present invention, and the present invention may be practiced without the specific details. In other instances, well known features are omitted or simplified in order not to obscure the present invention.

Various operations will be described as multiple discrete steps performed in turn in a manner that is most helpful in understanding the present invention. However, the order of description should not be construed as to imply that these operations are necessarily performed in the order they are presented, or even order dependent. Lastly, repeated usage of the phrase “in one embodiment” does not necessarily refer to the same embodiment, although it may.

Referring now to **Figure 1**, wherein a block diagram illustrating an overview of the present invention, in accordance with one embodiment is shown. As illustrated, video source device **102** and video sink device **104** are coupled to each other via digital video link **106**. Video source device **102** includes video source application **108** and video hardware interface **110**. Video source application **108** generates and provides video content to video hardware interface **110**, which in turn ciphers video content and provides the video content in a ciphered form to video sink device **104** through digital video link **106** as disclosed in the aforementioned parent applications, thereby protecting video contents. Additionally, video source application **108** and video hardware interface **110** exchange various status and

control information, including in particular status information about the link between video hardware interface **110** and video sink device **104**, and secret values employed by video hardware interface **110** to cipher video content as disclosed in the parent applications. In accordance with the present invention, video source application **108** and video hardware interface **110** are equipped to be able to jointly practice a symmetric ciphering/deciphering process. As a result, at least status and secret values may be provided from video hardware interface **110** to video source application **108** in a protected manner, maintaining protection to the video content being distributed to video sink device **104**.

Except for the teachings of the present invention incorporated, to be described more fully below, video source application **108** is intended to represent a broad range of video source applications known in the art, while video hardware interface **110** is substantially constituted as disclosed in the parent applications. As will be readily apparent from those skilled in the art, the present invention advantageously allows the same hardware resources of video hardware interface **110** to be used to protect the exchanges with video source application **108** as well as protecting the video content transmitted to video sink device **104**.

As disclosed in the parent applications, examples of video source device **102** includes but not limited to computers of all sizes (from palm size device to desktop device, and beyond), set-up boxes, or DVD players, whereas examples of video sink devices include but not limited to CRT monitors, flat panel displays or television sets. As to digital video link **106**, it may be implemented in any one of a number of mechanical and electrical forms, as long as they are consistent with the operating requirement (i.e. speed, bit rate and so forth), and a mechanism (which may be in hardware or through protocol) is provided to allow control information to be exchanged between video source and sink devices **102** and **104**.

Before proceeding to further described the present invention, while for ease of understanding, video source application **108** is shown to be interacting with video hardware interface **110** "directly", those skilled in the art will appreciate that typically video hardware interface **110** has an associated driver to insulate the hardware specifics from the interacting software, such as video source application **108** in this case. Accordingly, in most embodiments, video source application **108** interacts with video hardware interface **110** through its associated driver.

Figures 2a-2b illustrate two overviews of the symmetric ciphering/deciphering process based method for facilitating exchanges of status and control information between video source application **108** and video hardware interface **110**, in accordance with two embodiments. **Fig. 2a** is an embodiment particularly suitable for exchanges involving status and control information of short bit lengths, such as on/off status, whereas **Fig. 2b** is an embodiment particular suitable for exchanges involving status and control information of longer bit lengths, such as the secret values employed by video hardware interface **110** to cipher video contents. What constitutes short or longer bit length is application dependent. As between video hardware interface **110** and video sink device **104**, video source application **108** and video hardware interface **110** are assumed to have each been provided with an array of private "cryptographic" keys and a complementary identifier by a certification authority. In one embodiment, each of video source application **108** and video hardware interface **104** is pre-provided with an array of 40 56-bit private "cryptographic" keys by the certification authority. Cn is a 64-bit random number, and the keys are 56-bit long. For more information on the above described authentication process, see co-pending U.S. Patent Application, serial number 09/275,722, filed on March 24, 1999, entitled Method and Apparatus for the

Generation of Cryptographic Keys, having common assignee with the present application.

As illustrated in **Fig. 2a**, whenever a need occurs for video source application to retrieve a status of the short bit length type, video source application **108** first generates and provides a basis value to the symmetric ciphering/deciphering process to sink hardware interface **110**. For the illustrated embodiment, the basis value is a random number (Cn). Cn may be generated in any one of a number of techniques known in the art. Additionally, video source application **108** also provides a key selection value (Ck_{sv}) to video hardware interface **110**. Further, for the illustrated embodiment, which is an embodiment where the same hardware resources of video hardware interface **110** are used to satisfy video source application's request for status and control information of the short or long bit length type, video source application **108** also provides a mode indicator (C_{mode}) to video hardware interface **110** to denote the type of status and control information being requested. These parameters, C_n, Ck_{sv}, and C_{mode} may be provided via one or more "packets", as well as in conjunction with other information.

In response, video hardware interface **110** generates an authentication key K_u' based on its provided array of private "cryptographic" keys Dkeys and the selection key Ck_{sv} provided by video source application **108**. Video hardware interface **110** then generates the verification key K_p' based on the provided basis value C_n, the generated authentication key K_u', the status to be returned, and the selection key Bk_{sv} it was provided by video sink device **104** for use to protectively provide video contents in a ciphered form to video sink device **104** based on a symmetric cipher/deciphering process (see parent application for further detail).

Upon generating K_p', for the illustrated embodiment, video hardware interface **110** returns the requested status along with K_p'. In one embodiment, the two values

are concatenated together (S'), and returned at the same time. In alternate embodiments, it may be returned separately. Additionally, for the illustrated embodiment, video hardware interface **110** also returns Bk_{sv} and Dk_{sv} to video source application **108**.

Over on the video source application side, upon receipt of S', Bk_{sv} and Dk_{sv}, video source application **108** independently generates its own copy of K_u based on its array of pre-provided private "cryptographic" keys Ckeys, and Dk_{sv}. Next, video source application **108** independently generates its own copy of K_p based on C_n, the returned status, and Bk_{sv}. Then, video source application **108** compares its independently generated K_p with the received K_p' to determine if it should trust the status provided (when K_p=K_p') or distrust the status provided (when K_p=/=K_p').

Referring now to **Fig. 2b**, in like manner, whenever a need occurs for video source application to retrieve a control information of the longer bit length type, such as the aforementioned secret value, video source application **108** also first generates and provides a basis value to the symmetric ciphering/deciphering process to sink hardware interface **110**. Again, in one embodiment, the basis value is a random number (Cn), and it may be generated in any one of a number of techniques known in the art. Additionally, video source application **108** also provides a key selection value (Ck_{sv}) to video hardware interface **110**. Further, similar to the embodiment of **Fig. 2a**, where the same hardware resources of video hardware interface **110** are used to satisfy video source application's request for status and control information of the short or long bit length type, video source application **108** also provides a mode indicator (C_{mode}) to video hardware interface **110** to denote the type of status and control information being requested. As before, these parameters, C_n, Ck_{sv}, and C_{mode} may be provided via one or more "packets", as well as in conjunction with other information.

In response, video hardware interface **110** generates an authentication key K_u' based on its provided array of private "cryptographic" keys Dkeys and the selection key Ck_{sv} provided by video source application **108**. Video hardware interface **110** then generates a cryptographic key K_e' using K_u' and the provided basis value C_n.

Upon generating K_e', video hardware interface **110** ciphers the requested control information, e.g. secret value M_0', using K_e'. Video hardware interface **110** then returns M_0' in a ciphered form (M') to video source application **108**. Additionally, for the illustrated embodiment, video hardware interface **110** also returns Dk_{sv} to video source application **108**.

Over on the video source application side, upon receipt of M' and Dk_{sv}, video source application **108** independently generates its own copy of K_u based on Ckeys and Dk_{sv}. Next, video source application **108** independently generates its own copy of K_e based on C_n and K_u. Then, video source application **108** deciphers M', recovering M_0' using K_e.

Figures 3a-3b illustrate the symmetric ciphering/deciphering processes of **Fig.2a-2b** in further detail, in accordance with one embodiment each. As illustrated in **Fig. 3a**, for the exchange of status and control information of short bit length, video hardware interface **110** first generates the authentication key K_u' by summing its pre-provided private "cryptographic" keys Dkeys over the provided selection key Ck_{sv} from video source application **108**. Upon generation of the authentication key K_u', video hardware interface **110** generates a first intermediate key K_1', ciphering the least significant 40 bits (LSB40) of the provided basis value C_n by applying a one way function to it, using K_u'. For the illustrated embodiment, the same one way function is used for the exchange of status and control information of

both short and longer bit length type. The one way function is applied in a first mode, also referred to as the A-mode, in accordance with the value of C_{mode}. Next, video hardware interface **110** generates a second intermediate key K_2' by applying the same one way function (under the same mode) to the selection key BK_{sv} provided by video sink device **104**, using K_1'. Finally, video hardware interface **110** generates the verification key K_p' by applying the same one way function (under the same mode) to the status concatenated with most significant 24 bits (MSB24) of the provided basis value C_n, using K_2'.

Over on the video source application side, upon receipt of S', Dk_{sv}, and BK_{sv}, video source application **108** first independently generates its own copy of the authentication key K_u by summing its selection keys Ckeys over Dk_{sv}. Upon generation of the authentication key K_u, video source application **108** independently generates its own copy of the first intermediate key K_1 by applying a similar one way function to the least significant 40 bits (LSB40) of the basis value C_n provided to video hardware interface **110**, using K_u. Video source application **108** also uses the same one way function to facilitate the exchange of status and control information of both short and longer bit length type. Thus, the common one way function is applied in the earlier described first mode, also referred to as the A-mode, in accordance with the value of C_{mode}. Next, video source application **108** independently generates its own copy of the second intermediate key K_2 by applying the same one way function (under the same mode) to the selection key BK_{sv}, using K_1. Finally, video source application **108** independently generates its own copy of K_p by applying the same one way function (under the same mode) to the status concatenated with the most significant 24 bits (MSB24) of the basis value C_n, using K_2.

Fig. 3b illustrates the embodiment for handling the exchange of status and control information of longer bit length, video hardware interface **110** first generates the authentication key K_u' by summing its selected one of the pre-provided private "cryptographic" keys over the provided selection key from video source application **108**. Upon generation of the authentication key K_u', video hardware interface **110** generates another intermediate key K_4' by applying a one way function to the least significant 40 bits (LSB40) of the provided basis value C_n, using K_u'. For the illustrated embodiment, the same one way function is used for the exchange of status and control information of both short and longer bit length type. The one way function is applied in a second mode, also referred to as the B-mode, in accordance with the value of C_{mode}. Next, video hardware interface **110** generates K_e', the ciphering key, by applying the same one way function (under the same mode) to the most significant 24 bits (MSB24) of the provided basis value C_n, using K_4'.

Over on the video source application side, upon receipt of M' and Dk_{sv}, video source application **108** first independently generates its own copy of the authentication key K_u by summing its array of private "cryptographic" keys Ckeys over Dk_{sv}. Upon generation of the authentication key K_u, video source application **108** independently generates its own copy of intermediate key K_4 by applying a similar one way function to the least significant 40 bits (LSB40) of the basis value C_n, using K_u. Video source application **108** also uses the same one way function to facilitate the exchange of status and control information of both short and longer bit length type. Thus, the common one way function is applied in the earlier described second mode, also referred to as the B-mode, in accordance with the value of C_{mode}. Next, video source application **108** independently generates its own copy of K_p, the deciphering key, by applying the same one way function (under the same mode) to the most significant 24 bits (MSB24) of the basis value C_n, using K_1.

In one embodiment, K_1 and K_4 are generated only by video source application **108**, once per "session", using highly protected Ckeys, and stored in the application for later use for the remainder of the session. In other words, compromise of K1 or K4 allows "attack" for only one session (compromise of Ckeys would allow "attack" for unlimited number of sessions). This approach has the following advantages. Since Dk_{sv} is a constant, video source application **108** can fix the least significant 40 bits of C_n, and change only the most significant 24 bits of C_n for different status and information requests, thereby allowing video source application **108** to rerun the protocol for different requests at the computation of K_1 and K_4 and speed up the transfer of these information.

Figures 4a-4c illustrate a one-way function suitable for use to practice the symmetric ciphering/deciphering process of **Fig. 3a-3b**, in accordance with one embodiment. As illustrated in **Fig. 4a**, the one way function **800** includes a number of linear feedback shift registers (LFSRs) **802** and combiner function **804**, coupled to each other as shown. LFSRs **802** and combiner function **804** are collectively initialized with the appropriate keys and data values, depending the mode of operation C_{mode}. During operation, the values are successively shifted through LFSRs **802**. Selective outputs are taken from LFSRs **802**, and combiner function **804** is used to combine the selective outputs to generate the desired outputs.

In one embodiment, four LFSRs of different lengths are employed. Three sets of outputs are taken from the four LFSRs. The polynomials represented by the LFSR and the bit positions of the three sets of LFSR outputs are given by the table to follow:

LFSR	Polynomial	Combining Function Taps		
		0	1	2
3	$x^{27} + x^{24} + x^{21} + x^{17} + x^{13} + x^{8} + 1$	8	17	26
2	$x^{26} + x^{23} + x^{18} + x^{15} + x^{12} + x^{8} + 1$	8	16	25
1	$x^{24} + x^{21} + x^{18} + x^{14} + x^{10} + x^{7} + 1$	7	15	23
0	$x^{23} + x^{20} + x^{16} + x^{12} + x^{9} + x^{6} + 1$	7	14	22

The initialization of the LFSRs and the combiner function, more specifically, the shuffling network of the combiner function, is in accordance with the following table.

	Bit Field	One Way-A Initial Value	One Way-B Initial Value
LFSR3	[26:22]	Data [39:35]	Data[34:30]
	[21]	inverse of LFSR3 initialization bit [9]	inverse of LFSR3 initialization bit [9]
	[20:14]	Data[34:28]	Data[29:23]
	[13:0]	Key[55:42]	Key[48:35]
LFSR2	[25:22]	Data[27:24]	Data[22:19]
	[21]	inverse of LFSR2 initialization bit [8]	inverse of LFSR2 initialization bit [8]
	[20:14]	Data[23:17]	data[18:12]
	[13:0]	Key[41:28]	Key[34:21]
LFSR1	[23:19]	Data[16:12]	Data[11:7]

	[18]	inverse of LFSR1 initialization bit [5]	inverse of LFSR1 initialization bit [5]
	[17:14]	Data[11:8]	Data[6:3]
	[13:0]	Key[27:14]	Key[20:7]
LFSR0	[22:20]	Data[7:5]	Data[2:0]
	[19]	inverse of LFSR0 initialization bit [10]	inverse of LFSR0 initialization bit [10]
	[18:14]	Data[4:0]	Data[39:35]
	[13:7]	Key[13:7]	Key[6:0]
	[6:0]	Key[6:0]	Key[55:49]
Shuffle	Register A	0	0
Network	Register B	1	1

Data are LSB40(C_n), BK_{sv} and MSB24(C_n), whereas Keys are K_u, K_1, K_2 and K_4.

The combined result is generated from the third set of LFSR outputs, using the first and second set of LFSR outputs as data and control inputs respectively to combiner function **804**. The third set of LFSR outputs are combined into a single bit.

Fig. 4b illustrates combiner function **804** in further detail, in accordance with one embodiment. As illustrated, combiner function **804** includes shuffle network **806** and XOR **808a-808b**, serially coupled to each other and LFSRs **802** as shown. For the illustrated embodiment, shuffle network **806** includes four binary shuffle units **810a-810d** serially coupled to each other, with first and last binary shuffle units **810a** and **810d** coupled to XOR **808a** and **808b** respectively. XOR **808a** takes the first

group of LFSR outputs and combined them as a single bit input for shuffle network **806**. Binary shuffle units **810a-810d** serially propagate and shuffle the output of XOR **808a**. The second group of LFSR outputs are used to control the shuffling at corresponding ones of binary shuffle units **810a-810d**. XOR **808b** combines the third set of LFSR outputs with the output of last binary shuffle unit **810d**.

Fig. 4c illustrates one binary shuffle unit **810*** (where * is one of **a-d**) in further detail, in accordance with one embodiment. Each binary shuffle unit **810*** includes two flip-flops **812a** and **812b**, and a number of selectors **814a-814c**, coupled to each other as shown. Flip-flops **812a** and **812b** are used to store two state values (A, B). Each selector **814a**, **814b** or **814c** receives a corresponding one of the second group of LFSR outputs as its control signal. Selector **814a-814b** also each receives the output of XOR **808a** or an immediately preceding binary shuffle unit **810*** as input. Selector **814a-814b** are coupled to flip-flops **812a-812b** to output one of the two stored state values and to shuffle as well as modify the stored values in accordance with the state of the select signal. More specifically, for the illustrated embodiment, if the stored state values are (A, B), and the input and select values are (D, S), binary shuffle unit **810*** outputs A, and stores (B, D) if the value of S is "0". Binary shuffle unit **810*** outputs B, and stores (D, A) if the value of S is "1".

In one embodiment, once the data values are loaded into the registers and the shuffle networks, the one-way function is clocked for 32 clocks to mix the data and key bits. During this warm up period, the 32 output bits are discarded. As a result, the initial output stream is a non-linear function of many key and data bits. In alternate embodiments, depending on the desired robustness level, the present invention may be practiced with shorter or longer warm up period.

Those skilled in the art will appreciate that this one way function substantially parallel one embodiment of the one way function disclosed in the parent applications for the cipher employed by video hardware interface **110** to cipher video content to be transmitted to video sink device **104**. Accordingly, video hardware interface **110** may employ the same one way function to facilitate exchange of status and control information with video source application **108** in a protected manner, as well as to cipher video content for video sink device **104**.

Accordingly, a novel method and apparatus for ciphering and deciphering video content to protect the video content from unauthorized copying during transmission has been described.

Epilogue

From the foregoing description, those skilled in the art will recognize that many other variations of the present invention are possible. Thus, the present invention is not limited by the details described, instead, the present invention can be practiced with modifications and alterations within the spirit and scope of the appended claims.